Home Work 2

on Markov Decision Processes, Dynamic Programming

Srivatsava Kesanupalli MT18054

Question 2. Program to find state-value function and generate the values as in the fig 3.2 [Sutton & Barto]

Steps involved:

- A rectangular grid world representation of a simple finite MDP
- There are four possible actions north, east, west, south
- Actions that take of the agent from the grid will result in a reward of -1.0
- Other actions which include, the agent going off the grid would result in a reward of 0
- If the agent moves from state A i.e., from [0,1] to A' [4,1] — reward is +10
- If the agent moves from state B i.e., from [0,3] to B' [2,3] — reward is +5

Our goal is to find the state-value function $v_\pi(s)$ given by,

$$\sum \pi(s \mid a) \sum p(s', r \mid s, a)[r + \gamma v_\pi(s')] \ \forall \text{ s } \boldsymbol{\varepsilon} \text{ S}$$

Question 4. Program to find optimal state-value function and generate the values as in the fig 3.5 [Sutton & Barto]

Steps involved:

- A rectangular grid world representation of a simple finite MDP
- There are four possible actions north, east, west, south
- Actions that take of the agent from the grid will result in a reward of -1.0

- Other actions which include, the agent going off the grid would result in a reward of 0
- If the agent moves from state A i.e., from [0,1] to A' [4,1] — reward is +10
- If the agent moves from state B i.e., from [0,3] to B' [2,3] — reward is +5

Our goal is to find the optimal state-value function $v_*(s)$ given by,

$$max_a \sum p(s', r \mid s, a)[r + \gamma v_\star(s')]$$

Question 6. Policy Iteration and Value Iteration to solve the Gridworld example 4.1

Steps involved:

- The grid contains non-terminal state S = { 1, 2, 3, 4, …, 14}
- The possible actions are north, east, west, and south
- This is an episodic, undiscounted task. The actions take place with equal probability

Policy Iteration and Value Iteration as given in the text book involve, computing the value functions v_k iteratively. The final estimate is what we call v_π

Policy Iteration involves:

- Policy Evaluation & Policy Improvement

Policy Evaluation

• Find new value state and the difference between the new state and old state should be less than a value to terminate

Policy Improvement

• If the old action is same as the current action which is found by arg_max of the value functions with each action, we terminate the iterative procedure.

*** If the actions oscillate without no new action being selected, we check for such occurrence and include the condition is termination. This is a practical fix to the claimed bug in the text book

Value Iteration involves:

- Estimating the max of all the new state values and selecting it as the state value for the next iteration. This procedure is repeated until, we have a termination i.e., the difference between the states is less than a very small threshold